

Beijing, China, January 18-21, 2000

Agenda Item : AH04 + AH08
Source : Nortel Networks
Title : Compressed Mode by Puncturing
Document for : Decision

1. Introduction

In WG1#9, introduction of compressed mode by puncturing method has been discussed. A proposal from Nokia has been presented. This solution consisted in two main ideas. The first idea was to perform the supplementary puncturing in the normal rate matching block, over the whole Transmission Time Interval of the Transport Channel considered. The second idea was to modify Radio Frame Segmentation block to put the right number of bits in each frame according to the fact that it is compressed or not.

It seems that performing the supplementary rate matching over the whole TTI provides best performance since it leads to an optimal distance between the bits punctured or repeated, due to the usage of the currently specified rate matching algorithm.

However, it was felt that the changes of the following blocks of the multiplexing chain, in particular Radio Frame Segmentation block, was requiring to many modifications.

In this document, a follow up of the ideas in R1-99j04 is presented. The solution is characterised by the following :

All supplementary puncturing is performed in the rate matching block to obtain best distance.

In addition some bits marked with a fourth value (on top of 0, 1 and DTX indication) are inserted in positions that will go to the compressed frames. These bits will be removed in the Physical Channel Mapping step to create the gap needed in the compressed frames.

In order to avoid running a supplementary algorithm, the insertion of these bits marked p is done during the normal rate matching algorithm, by using information from first channel interleaver to know in which position bits p are to be inserted.

The calculations of the amount of rate matching needed take into account the number of bits to be "rate matched" for normal rate matching, plus the number of bits p to insert. More precisely, a number of bits equal to the number of bits corresponding to the gaps to create in the TTI are additionally punctured. In this way, the total number of bits after the rate matching block is the same as it would have been in normal mode but bits marked p have been inserted to represent the gap to create.

Having the number of bits output from the rate matching block being equal in compressed mode and normal mode allows to make no additional modification for compressed mode in the other block of the chain. Only Physical Channel Mapping has to eliminate the bits marked p, when it creates the gap for compressed mode.

2. Modification of notations

In order to let the introduction of proposed method be clearer, some modifications of the notations have been made. They also clarify the case of normal mode, in particular for Turbo Codes. However, they do not modify the algorithms currently specified, neither of course the results of the output bits flows. These modifications consist in the following.

Use arrays to identify the parameters for the Y and Y' streams of Turbo Codes: e_{ini}, e_{plus}, e_{minus} are arrays of size one for Convolutional Codes, and arrays of size two for Turbo Codes. Thus these parameters can be directly initialised with the right value instead of using the notation "a".

The range of the index m in the rate matching loop is modified to show that in the case of convolutional codes, and Turbo Codes with repetition, the rate matching applies to the flow x_{1ik} (referring to Figure 6, and 8), and that in the case of Turbo Codes, it applies to flow x_{2ik}, x_{3ik} (referring to Figures 7 and 9).

3. Introduction of new method

For the introduction of new method, two main changes are done.

The first one deals with calculation of the amount of bits to puncture or repeat in the rate matching, and the calculation of the number of bits p per frame and per TrCH corresponding to the gap to create in the compressed frames.

The second one deals with the insertion of the bits p in the right position while performing the rate matching.

3.1 Calculations for rate matching parameters in downlink

First, the number of bits corresponding to the gaps per frame in the TTI considered is calculated: $N_{TGL}[k]$.

Then the number of bits which correspond to the gap for each TrCh in each frame of the TTI is calculated. In order to respect the balancing obtained by the static rate matching factors, formula (1) with Z variable is used, replacing $N_{data,j}$ by $N_{TGL}[k]$.

The number of bits corresponding to the gaps in the TTI is calculated for each TrCh. The sum of this supplementary bits to be punctured and of the amount of normal rate matching is then used to run the current rate matching algorithm on the TTI.

It must be noted that for Turbo Codes the amount of bits to be rate matched is equilibrated between Parity 1 and Parity 2 flows by using alternatively up and down rounding.

3.2 Insertion of the bits p

While running the rate matching algorithm on the input bit flow, the frame to which each position will go is calculated using information of the first channel interleaver, and bits marked p are inserted in the positions corresponding to frames to be compressed.

**3GPP TSG RAN WG1 Meeting #10
Beijing, China, January 18 – 21, 2000**

Document R1-00-121

e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.212 CR 042

Current Version: **3.1.0**

GSM (AA.BB) or 3G (AA.BBB) specification number ↑

↑ CR number as allocated by MCC support team

For submission to: **TSG-RAN #7**
list expected approval meeting # here ↑

for approval
for information

strategic
non-strategic (for SMG use only)

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM ME UTRAN / Radio Core Network
(at least one should be marked with an X)

Source: Nortel Networks **Date:** 2000-01-14

Subject: DL compressed mode by puncturing

Work item:

Category:	F Correction	<input type="checkbox"/>	Release:	Phase 2	<input type="checkbox"/>
	A Corresponds to a correction in an earlier release	<input type="checkbox"/>		Release 96	<input type="checkbox"/>
(only one category shall be marked with an X)	B Addition of feature	<input type="checkbox"/>		Release 97	<input type="checkbox"/>
	C Functional modification of feature	<input checked="" type="checkbox"/>		Release 98	<input type="checkbox"/>
	D Editorial modification	<input type="checkbox"/>		Release 99	<input checked="" type="checkbox"/>
				Release 00	<input type="checkbox"/>

Reason for change: Introduce a new method to determine the position of the bits to be removed to create the transmission gap for compressed frames, while leading to minimal changes to the current multiplexing chain.

Clauses affected: 4.2.7

Other specs affected:	Other 3G core specifications	<input type="checkbox"/>	→ List of CRs:	
	Other GSM core specifications	<input type="checkbox"/>	→ List of CRs:	
	MS test specifications	<input type="checkbox"/>	→ List of CRs:	
	BSS test specifications	<input type="checkbox"/>	→ List of CRs:	
	O&M specifications	<input type="checkbox"/>	→ List of CRs:	

Other comments:



<----- double-click here for help and instructions on how to create a CR.

4.2.7 Rate matching

Rate matching means that bits on a transport channel are repeated or punctured. Higher layers assign a rate-matching attribute for each transport channel. This attribute is semi-static and can only be changed through higher layer signalling. The rate-matching attribute is used when the number of bits to be repeated or punctured is calculated.

The number of bits on a transport channel can vary between different transmission time intervals. In the downlink the transmission is interrupted if the number of bits is lower than maximum. When the number of bits between different transmission time intervals in uplink is changed, bits are repeated or punctured to ensure that the total bit rate after TrCH multiplexing is identical to the total channel bit rate of the allocated dedicated physical channels.

If no bits are input to the rate matching for all TrCHs within a CCTrCH, the rate matching shall output no bits for all TrCHs within the CCTrCH and no uplink DPDCH will be selected in the case of uplink rate matching.

Notation used in section 4.2.7 and subsections:

N_{ij} : For uplink: Number of bits in a radio frame before rate matching on TrCH i with transport format combination j .

For downlink : An intermediate calculation variable (not an integer but a multiple of 1/8).

N_{il}^{TTI} : Number of bits in a transmission time interval before rate matching on TrCH i with transport format l .
Used in downlink only.

ΔN_{ij} : For uplink: If positive - number of bits that should be repeated in each radio frame on TrCH i with transport format combination j .

If negative - number of bits that should be punctured in each radio frame on TrCH i with transport format combination j .

For downlink : An intermediate calculation variable (not an integer but a multiple of 1/8).

ΔN_{il}^{TTI} : If positive - number of bits to be repeated in each transmission time interval on TrCH i with transport format j .

If negative - number of bits to be punctured in each transmission time interval on TrCH i with transport format j .

Used in downlink only.

$\Delta N_{cm,ij}^{TTI}$: Negative or null: number of bits to be removed in one Transmission Time Interval, to create the required gaps in the compressed frames of this TTI, in case of compressed mode for TrCH i with transport format j .

Used in downlink only.

$\Delta N_{cm,ij}[k]$, $k=0$ to $F_i - 1$: Negative or null: number of bits, in each frame of the TTI corresponding to the gap for compressed mode in this frame, for TrCH i with transport format j . The value will be null for the uncompressed frames.

Used in downlink only.

$N_{TGL}[k]$, $k=0$ to $F_i - 1$: Positive or null: number of bits in each frame of the TTI of a Transport Channel, corresponding to the gap for compressed mode in a radio frame for the CCTrCh.

RM_i : Semi-static rate matching attribute for transport channel i . Signalled from higher layers.

PL : Puncturing limit for uplink. This value limits the amount of puncturing that can be applied in order to avoid multicode or to enable the use of a higher spreading factor. Signalled from higher layers.

$N_{data,j}$: Total number of bits that are available for the CCTrCH in a radio frame with transport format combination j .

- I : Number of TrCHs in the CCTrCH.
- Z_{ij} : Intermediate calculation variable.
- F_i : Number of radio frames in the transmission time interval of TrCH i .
- n_i : Radio frame number in the transmission time interval of TrCH i ($0 \leq n_i < F_i$).
- q : Average puncturing or repetition distance (normalised to only show the remaining rate matching on top of an integer number of repetitions). Used in uplink only.
- $I_F(n_i)$: The inverse interleaving function of the 1st interleaver (note that the inverse interleaving function is identical to the interleaving function itself for the 1st interleaver). Used in uplink only.
- $S(n_i)$: The shift of the puncturing or repetition pattern for radio frame n_i . Used in uplink only.
- $TF_i(j)$: Transport format of TrCH i for the transport format combination j .
- $TFS(i)$: The set of transport format indexes l for TrCH i .
- $TFCS$: The set of transport format combination indexes j .

$e_{ini}[x], x=1$ to 2 :

Initial value of variable e in the rate matching pattern determination algorithm of section 4.2.7.5.

$e_{ini}[1]$: initial value for convolutional codes and for turbo codes Parity-1 bits

$e_{ini}[2]$: initial value for turbo codes Parity-2 bits

$e_{plus}[x], x=1$ to 2 :

Increment of variable e in the rate matching pattern determination algorithm of section 4.2.7.5.

$e_{plus}[1]$: Increment of variable e in the rate matching for convolutional codes and for turbo codes Parity-1 bits

$e_{plus}[2]$: Increment of variable e in the rate matching for turbo codes Parity-2 bits

$e_{minus}[x], x=1$ to 2 :

Decrement of variable e in the rate matching pattern determination algorithm of section 4.2.7.5.

$e_{minus}[1]$: Decrement of variable e in the rate matching for convolutional codes and for turbo codes Parity-1 bits

$e_{minus}[2]$: Decrement of variable e in the rate matching for turbo codes Parity-2 bits

$\Delta N_j[x], x=1$ to 2 :

Number of bits to initialise parameters for the rate matching pattern determination algorithm of section 4.2.7.5.

$\Delta N_j[1]$: Parameter for convolutional codes and Turbo Codes Parity-1 bits

$\Delta N_j[2]$: Parameter for convolutional codes and Turbo Codes Parity-2 bits

b : Indicates systematic and parity bits

$b=1$: Systematic bit. $X(t)$ in section 4.2.3.2.1.

$b=2$: 1st parity bit (from the upper Turbo constituent encoder). $Y(t)$ in section 4.2.3.2.1.

$b=3$: 2nd parity bit (from the lower Turbo constituent encoder). $Y'(t)$ in section 4.2.3.2.1.

The * (star) notation is used to replace an index x when the indexed variable X_x does not depend on the index x . In the left wing of an assignment the meaning is that " $X_* = Y$ " is equivalent to "**for all x do $X_x = Y$** ". In the right wing of an assignment, the meaning is that " $Y = X_*$ " is equivalent to "**take any x and do $Y = X_x$** "

The following relations, defined for all TFC j , are used when calculating the rate matching parameters:

$$Z_{0,j} = 0$$

$$Z_{ij} = \left[\frac{\sum_{m=1}^i RM_m \cdot N_{mj}}{\sum_{m=1}^I RM_m \cdot N_{mj}} \cdot N_{data,j} \right] \text{ for all } i = 1 \dots I \quad (1)$$

$$\Delta N_{ij} = Z_{ij} - Z_{i-1,j} - N_{ij} \quad \text{for all } i = 1 \dots I$$

4.2.7.1 Determination of rate matching parameters in uplink

4.2.7.1.1 Determination of SF and number of PhCHs needed

In uplink, puncturing can be applied to match the CCTrCH bit rate to the PhCH bit rate. The bit rate of the PhCH(s) is limited by the UE capability and restrictions imposed by UTRAN, through limitations on the PhCH spreading factor. The maximum amount of puncturing that can be applied is signalled from higher layers and denoted by PL . The number of available bits in the radio frames for all possible spreading factors is given in [2]. Denote these values by N_{256} , N_{128} , N_{64} , N_{32} , N_{16} , N_8 , and N_4 , where the index refers to the spreading factor. The possible values of N_{data} then are $\{N_{256}, N_{128}, N_{64}, N_{32}, N_{16}, N_8, N_4, 2N_4, 3N_4, 4N_4, 5N_4, 6N_4\}$. Depending on the UE capability and the restrictions from UTRAN, the allowed set of N_{data} , denoted SET0, can be a subset of $\{N_{256}, N_{128}, N_{64}, N_{32}, N_{16}, N_8, N_4, 2N_4, 3N_4, 4N_4, 5N_4, 6N_4\}$. $N_{data,j}$ for the transport format combination j is determined by executing the following algorithm:

$$\text{SET1} = \{ N_{data} \text{ in SET0 such that } N_{data} - \sum_{x=1}^I \frac{RM_x}{\min_{1 \leq y \leq I} \{RM_y\}} \cdot N_{x,j} \text{ is non negative} \}$$

If SET1 is not empty and the smallest element of SET1 requires just one PhCH then

$$N_{data,j} = \min \text{SET1}$$

else

$$\text{SET2} = \{ N_{data} \text{ in SET0 such that } N_{data} - PL \cdot \sum_{x=1}^I \frac{RM_x}{\min_{1 \leq y \leq I} \{RM_y\}} \cdot N_{x,j} \text{ is non negative} \}$$

Sort SET2 in ascending order

$$N_{data} = \min \text{SET2}$$

While N_{data} is not the max of SET2 and the follower of N_{data} requires no additional PhCH do

$$N_{data} = \text{follower of } N_{data} \text{ in SET2}$$

End while

$$N_{data,j} = N_{data}$$

End if

4.2.7.1.2 Determination of parameters needed for calculating the rate matching pattern

The number of bits to be repeated or punctured, DN_{ij} , within one radio frame for each TrCH i is calculated with equation 1 for all possible transport format combinations j and selected every radio frame. $N_{data,j}$ is given from section 4.2.7.1.1. In compressed mode $N_{data,j}$ is replaced by $N_{data,j}^{cm}$ in Equation 1. $N_{data,j}^{cm}$ is given from the following relation:

$$N_{data,j}^{cm} = 2N_{data,j} - 2N_{TGL}, \text{ for compressed mode by spreading factor reduction}$$

$$N_{data,j}^{cm} = N_{data,j} - N_{TGL}, \text{ for compressed mode by higher layer scheduling}$$

$$N_{TGL} = \begin{cases} \frac{TGL}{15} N_{data,j}, & \text{if } N_{first} + TGL \leq 15 \\ \frac{15 - N_{first}}{15} N_{data,j}, & \text{in first frame if } N_{first} + TGL > 15 \\ \frac{TGL - (15 - N_{first})}{15} N_{data,j}, & \text{in second frame if } N_{first} + TGL > 15 \end{cases}$$

N_{first} and TGL are defined in section 4.4.

If $DN_{ij} = 0$ then the output data of the rate matching is the same as the input data and the rate matching algorithm of section 4.2.7.5 does not need to be executed.

If $DN_{ij} \neq 0$ the parameters listed in sections 4.2.7.1.2.1 and 4.2.7.1.2.2 shall be used for determining e_{ini} , e_{plus} , and e_{minus} (regardless if the radio frame is compressed or not).

4.2.7.1.2.1 Uncoded and convolutionally encoded TrCHs

$R = DN_{ij} \bmod N_{ij}$ -- note: in this context $DN_{ij} \bmod N_{ij}$ is in the range of 0 to $N_{ij}-1$ i.e. $-1 \bmod 10 = 9$.

if $R \neq 0$ and $2R \leq N_{ij}$

$$\text{then } q = \epsilon N_{ij} / R \hat{u}$$

else

$$q = \epsilon N_{ij} / (R - N_{ij}) \hat{u}$$

endif

-- note: q is a signed quantity.

if q is even

$$\text{then } q' = q + \gcd(|q|, F_i)/F_i \text{ -- where } \gcd(|q|, F_i) \text{ means greatest common divisor of } |q| \text{ and } F_i$$

-- note that q' is not an integer, but a multiple of 1/8

else

$$q' = q$$

endif

for $x = 0$ to F_i-1

$$S(I_F(|\lfloor x * q' \rfloor| \bmod F_i)) = (\lfloor \lfloor x * q' \rfloor \rfloor \bmod F_i)$$

end for

$$\Delta N_i = \Delta N_{i,j}$$

$$a=2$$

For each radio frame, the rate-matching pattern is calculated with the algorithm in section 4.2.7.5, where :

$$X_i = N_{i,j}, \text{ and}$$

$$e_{mi}[1] = (2a \cdot S(n_i) \cdot |\Delta N_i| + 1) \bmod 2a \cdot N_{ij}.$$

$$e_{plus}[1] = 2a \cdot N_{ij}$$

$$e_{minus}[1] = 2a \cdot |\Delta N_i|$$

puncturing for $\Delta N < 0$, repetition otherwise.

4.2.7.1.2.2 Turbo encoded TrCHs

If repetition is to be performed on turbo encoded TrCHs, i.e. $\Delta N_{i,j} > 0$, the parameters in section 4.2.7.1.2.1 are used.

If puncturing is to be performed, the parameters below shall be used. Index b is used to indicate systematic ($b=1$), 1st parity ($b=2$), and 2nd parity bit ($b=3$).

$$a=2 \text{ when } b=2$$

$$a=1 \text{ when } b=3$$

$$\Delta N_i = \begin{cases} \lfloor \Delta N_{i,j} / 2 \rfloor, & b = 2 \\ \lfloor \Delta N_{i,j} / 2 \rfloor, & b = 3 \end{cases}$$

$$\Delta N_i [1] = \lfloor \Delta N_{i,j} / 2 \rfloor$$

$$\Delta N_i [2] = \lceil \Delta N_{i,j} / 2 \rceil$$

For Parity-1 flow, calculate S1 as S in the following algorithm by setting ΔN_i to $\Delta N_i [1]$, and b to 2.

For Parity-2 flow, calculate S2 as S in following algorithm by setting ΔN_i to $\Delta N_i [2]$, and b to 3

$$X_i = \lfloor N_{i,j} / 3 \rfloor,$$

$$q = \lfloor X_i / \Delta N_i \rfloor$$

if($q \leq 2$)

for $x=0$ to F_i-1

$$S[\text{I}_F[(3x+b-1) \bmod F_i]] = x \bmod 2;$$

end for

else

if q is even

then $q' = q - \text{gcd}(q, F_i) / F_i$ -- where $\text{gcd}(q, F_i)$ means greatest common divisor of q and F_i
-- note that q' is not an integer, but a multiple of 1/8

else $q' = q$

endif

for $x=0$ to $F_i - 1$

$$r = \lceil x * q' \rceil \bmod F_i;$$

$$S[\lfloor (3r+b-1) \bmod F_i \rfloor] = \lceil x \cdot q \rceil \div F_i;$$

endfor

endif

For each radio frame, the rate-matching pattern is calculated with the algorithm in section 4.2.7.5, where:

X_i is as above,

$$e_{ini}[1] = (\alpha_2 \cdot S1(n_i) \cdot DN_i[1] + X_i) \bmod (\alpha_2 \cdot X_i), \text{ if } e_{ini}[1] = 0 \text{ then } e_{ini}[1] = \alpha_2 \cdot X_i.$$

$$e_{plus}[1] = \alpha_2 \cdot X_i$$

$$e_{minus}[1] = \alpha_2 \cdot DN_i$$

$$e_{ini}[2] = (\alpha_2 \cdot S2(n_i) \cdot DN_i[2] + X_i) \bmod X_i, \text{ if } e_{ini}[2] = 0 \text{ then } e_{ini}[2] = X_i.$$

$$e_{plus}[2] = X_i$$

$$e_{minus}[2] = DN_i$$

4.2.7.2 Determination of rate matching parameters in downlink

For downlink $N_{data,j}$ does not depend on the transport format combination j . $N_{data,*}$ is given by the channelization code(s) assigned by higher layers.

When compressed mode by puncturing is used, the number of bits corresponding to the gap for TrCh i , in each frame of its TTI is noted $\Delta N_{cm,ij}[k]$ is calculated as follows.

For each frame k of the TTI, $N_{TGL}[k]$ is given by the relation:

$$N_{TGL} = \begin{cases} \frac{TGL}{15} N_{data,j}, & \text{if } N_{first} + TGL \leq 15 \\ \frac{15 - N_{first}}{15} N_{data,j}, & \text{in first frame if } N_{first} + TGL > 15 \\ \frac{TGL - (15 - N_{first})}{15} N_{data,j}, & \text{in second frame if } N_{first} + TGL > 15 \end{cases}$$

N_{first} and TGL are defined in section 4.4.

Note that $N_{TGL}[k] = 0$ if frame k is not compressed.

4.2.7.2.1 Determination of rate matching parameters for fixed positions of TrCHs

First an intermediate calculation variable $N_{i,*}$ is calculated for all transport channels i by the following formula:

$$N_{i,*} = \frac{1}{F_i} \cdot \max_{l \in TFS(i)} N_{i,l}^{TTI}$$

The computation of the $\Delta N_{i,l}^{TTI}$ parameters is then performed in for all TrCH i and all TF l by the following formula, where $\Delta N_{i,*}$ is derived from $N_{i,*}$ by the formula given at section 4.2.7:

$$\Delta N_{max} = F_i \cdot \Delta N_{i,*}$$

In normal mode, If $\Delta N_{max} = 0$ then, for TrCH i , the output data of the rate matching is the same as the input data and the rate matching algorithm of section 4.2.7.5 does not need to be executed. In this case we have :

$$\forall l \in TFS(i) \Delta N_{i,l}^{TTI} = 0$$

If $\Delta N_{max} \neq 0$ the parameters listed in sections 4.2.7.2.1.1 and 4.2.7.2.1.2 shall be used for determining e_{ini} , e_{plus} , and e_{minus} .

For compressed mode:

Let $\Delta N_{cm,max}[k]$ be the number of bits to eliminate on TrCH i to create the gap for compressed mode, in each frame k of the TTI, calculated for the TFC where the number of bits of the TrCh is maximum.

$\Delta N_{cm,max}[k]$ is calculated for TrCh i , for TFC max, for each frame k of the TTI using the formula (1) in 4.2.7, and replacing $N_{data,j}$ by $N_{TGI}[k]$.

The total number of bits $\Delta N_{cm,max}^{TTI}$ corresponding to the gaps for compressed mode for TrCh i in the TTI is calculated as:

$$\Delta N_{cm,max}^{TTI} = \sum_{k=0}^{F_i-1} \Delta N_{cm,max}[k]$$

In compressed mode, if $\Delta N_{max} = 0$ and $\Delta N_{cm,max}^{TTI} = 0$, then, for TrCH i , the output data of the rate matching is the same as the input data and the rate matching algorithm of section 4.2.7.5 does not need to be executed. If $\Delta N_{max} = 0$ and $\Delta N_{cm,max}^{TTI} > 0$, then, for TrCH i , the rate matching algorithm of section 4.2.7.5 needs to be executed.

4.2.7.2.1.1 Uncoded and convolutionally encoded TrCHs

For normal mode: $\Delta N_i = \Delta N_{max}$

For compressed mode: $\Delta N_i = \Delta N_{max} - \Delta N_{cm,max}^{TTI}$

$a=2$

$$N_{max} = \max_{l \in TFS(i)} N_{il}^{TTI}$$

For each transmission time interval of TrCH i with TF l , the rate-matching pattern is calculated with the algorithm in section 4.2.7.5. The following parameters are used as input:

$$X_i = N_{il}^{TTI}$$

$$e_{ini} = \lfloor e_{ini}[l] \rfloor = l$$

$$e_{plus} = a \cdot N_{max} - e_{plus}[l] = 2 \cdot N_{max}$$

$$e_{minus} = a \cdot \lfloor \Delta N_i \rfloor - e_{minus}[l] = 2 \cdot \lfloor \Delta N_i \rfloor$$

In normal mode, Puncturing if $\Delta N_i < 0$, repetition otherwise. The values of $\Delta N_{i,l}^{TTI}$ may be computed by counting repetitions or puncturing when the algorithm of section 4.2.7.5 is run.

In compressed mode, puncturing is performed if $\Delta N_i < 0$, or if $\Delta N_i = 0$ and $\Delta N_{cm,max}^{TTI} > 0$. Repetition is performed if $\Delta N_i > 0$.

4.2.7.2.1.2 Turbo encoded TrCHs

If repetition is to be performed on turbo encoded TrCHs, i.e. $\Delta N_{max} > 0$, the parameters in section 4.2.7.2.1.1 are used.

If puncturing is to be performed, the parameters below shall be used. Index b is used to indicate systematic ($b=1$), 1st parity ($b=2$), and 2nd parity bit ($b=3$).

$$a=2 \text{ when } b=2$$

$$a=1 \text{ when } b=3$$

The systematic bits indicated by $b=1$ shall not be punctured.

In Normal Mode

$$\Delta N_i [1] = \lfloor \Delta N_{max} / 2 \rfloor$$

$$\Delta N_i [2] = \lceil \Delta N_{max} / 2 \rceil$$

$$\Delta N_i = \begin{cases} \lfloor \Delta N_{max} / 2 \rfloor, & b = 2 \\ \lceil \Delta N_{max} / 2 \rceil, & b = 3 \end{cases}$$

In Compressed Mode

$$\Delta N_i [1] = \lfloor \Delta N_{max} / 2 \rfloor - \lceil \Delta N_{max}^{cm} / 2 \rceil$$

$$\Delta N_i [2] = \lceil \Delta N_{max} / 2 \rceil - \lfloor \Delta N_{max}^{cm} / 2 \rfloor$$

$$N_{max} = \max_{l \in TFS(i)} (N_{il}^{TTI} / 3)$$

For each transmission time interval of TrCH i with TF l , the rate-matching pattern is calculated with the algorithm in section 4.2.7.5. The following parameters are used as input:

$$X_i = N_{il}^{TTI} / 3$$

$$e_{ini} = N_{max} \quad e_{ini} [1] = N_{max}; \quad e_{ini} [2] = N_{max}$$

$$e_{plus} = a \cdot N_{max} \quad e_{plus} [1] = 2 N_{max}; \quad e_{plus} [2] = N_{max}$$

$$e_{minus} = a \cdot \lfloor \Delta N_i \rfloor \quad e_{minus} [1] = 2 \lfloor \Delta N_i \rfloor; \quad e_{minus} [2] = \lfloor \Delta N_i \rfloor$$

The values of $\Delta N_{i,l}^{TTI}$ may be computed by counting repetitions or puncturing when the algorithm of section 4.2.7.5 is run.

4.2.7.2.2 Determination of rate matching parameters for flexible positions of TrCHs

First an intermediate calculation variable N_{ij} is calculated for all transport channels i and all transport format combinations j by the following formula:

$$N_{i,j} = \frac{1}{F_i} \cdot N_{i,TF_i(j)}^{TTI}$$

Then rate matching ratios RF_i are calculated for each the transport channel i in order to minimise the number of DTX bits when the bit rate of the CCTrCH is maximum. The RF_i ratios are defined by the following formula:

$$RF_i = \frac{N_{data,*}}{\max_{j \in TFCS} \sum_{i=1}^{i=I} (RM_i \cdot N_{i,j})} \cdot RM_i$$

The computation of $\Delta N_{i,l}^{TTI}$ parameters is then performed in two phases. In a first phase, tentative temporary values of $\Delta N_{i,l}^{TTI}$ are computed, and in the second phase they are checked and corrected. The first phase, by use of the RF_i ratios, ensures that the number of DTX indication bits inserted is minimum when the CCTrCH bit rate is maximum, but it does not ensure that the maximum CCTrCH bit rate is not greater than $N_{data,*}$ per 10ms. The latter condition is ensured through the checking and possible corrections carried out in the second phase.

At the end of the second phase, the latest value of $\Delta N_{i,l}^{TTI}$ is the definitive value.

The first phase defines the tentative temporary $\Delta N_{i,l}^{TTI}$ for all transport channel i and any of its transport format l by use of the following formula:

$$\Delta N_{i,l}^{TTI} = F_i \cdot \left[\frac{RF_i \cdot N_{i,l}^{TTI}}{F_i} \right] - N_{i,l}^{TTI}$$

The second phase is defined by the following algorithm:

for all j in $TFCS$ do -- for all TFC

$$D = \sum_{i=1}^{i=I} \frac{N_{i,TF_i(j)}^{TTI} + \Delta N_{i,TF_i(j)}^{TTI}}{F_i} \quad \text{-- CCTrCH bit rate (bits per 10ms) for TFC } l$$

if $D > N_{data,*}$ then

for $i = 1$ to I do -- for all TrCH

$$\Delta N = F_i \cdot \Delta N_{i,j} \quad \text{-- } \Delta N_{i,j} \text{ is derived from } N_{i,j} \text{ by the formula given at section 4.2.7.}$$

if $\Delta N_{i,TF_i(j)}^{TTI} > \Delta N$ then

$$\Delta N_{i,TF_i(j)}^{TTI} = \Delta N$$

end-if

end-for

end-if

end-for

NOTE: The order in which the transport format combinations are checked does not change the final result.

For normal mode. If $\Delta N_{i,l}^{TTI} = 0$ then, for TrCH i at TF l , the output data of the rate matching is the same as the input data and the rate matching algorithm of section 4.2.7.5 does not need to be executed.

If $\Delta N_{i,l}^{TTI} \neq 0$ the parameters listed in sections 4.2.7.2.2.1 and 4.2.7.2.2.2 shall be used for determining e_{ini} , e_{plus} , and e_{minus} .

For compressed mode:

The number of bits for TrCh i corresponding to the gap for compressed mode in frame k of the TTI, noted $\Delta N_{cm,il}[k]$, is calculated for each frame k of the TTI using the formula (1) in 4.2.7, and replacing $N_{data,i}$ by $N_{TCL}[k]$.

The total number of bits $\Delta N_{cm,il}^{TTI}$ corresponding to the gaps for compressed mode for TrCh i in the whole TTI is calculated as:

$$\Delta N_{cm,il}^{TTI} = \sum_{k=0}^{F_i-1} \Delta N_{cm,il}[k]$$

If $\Delta N_{i,l}^{TTI} = 0$ and $\Delta N_{cm,il}^{TTI} = 0$, then, for TrCH i at TF l , the output data of the rate matching is the same as the input data and the rate matching algorithm of section 4.2.7.5 does not need to be executed.

Else the parameters listed in sections 4.2.7.2.2.1 and 4.2.7.2.2.2 shall be used for determining e_{mi} , e_{plus} , and e_{minus} .

4.2.7.2.2.1 Uncoded and convolutionally encoded TrCHs

In Normal Mode: $\Delta N_i = \Delta N_{il}^{TTI}$

In Compressed Mode: $\Delta N_i = \Delta N_{il}^{TTI} - \Delta N_{cm,il}^{TTI}$

$$\alpha = 2$$

For each transmission time interval of TrCH i with TF l , the rate-matching pattern is calculated with the algorithm in section 4.2.7.5. The following parameters are used as input:

$$X_i = N_{il}^{TTI}$$

$$e_{mi}[l] = 1 \cdot e_{mi} = 1$$

$$e_{plus}[l] = 2 \cdot N_{il}^{TTI} \cdot e_{plus} = \alpha \cdot N_{il}^{TTI}$$

$$e_{minus}[l] = 2 \cdot |\Delta N_i| \cdot e_{minus} = \alpha \cdot |\Delta N_i|$$

In Normal Mode : puncturing for $\Delta N_i < 0$, repetition otherwise.

In Compressed Mode: puncturing is performed if $\Delta N_i < 0$, or if $\Delta N_i = 0$ and $\Delta N_{cm,il}^{TTI} < 0$. Repetition is performed if $\Delta N_i > 0$.

4.2.7.2.2.2 Turbo encoded TrCHs

If repetition is to be performed on turbo encoded TrCHs, i.e. $\Delta N_{il}^{TTI} > 0$, the parameters in section 4.2.7.2.2.1 are used.

If puncturing is to be performed, the parameters below shall be used. Index b is used to indicate systematic ($b=1$), 1st parity ($b=2$), and 2nd parity bit ($b=3$).

$$\alpha = 2 \text{ when } b=2$$

$$\alpha = 1 \text{ when } b=3$$

The bits indicated by $b=1$ shall not be punctured.

In Normal Mode:

$$\Delta N_i = \begin{cases} \lfloor \Delta N_{il}^{TTI} / 2 \rfloor, & b = 2 \\ \lfloor \Delta N_{il}^{TTI} / 2 \rfloor, & b = 3 \end{cases}$$

$$\Delta N_i [1] = \lfloor \Delta N_{il}^{TTI} / 2 \rfloor$$

$$\Delta N_i [2] = \lceil \Delta N_{il}^{TTI} / 2 \rceil$$

In Compressed Mode:

$$\Delta N_i [1] = \lfloor \Delta N_{il}^{TTI} / 2 \rfloor - \lceil \Delta N_{cm,il}^{TTI} / 2 \rceil$$

$$\Delta N_i [2] = \lceil \Delta N_{il}^{TTI} / 2 \rceil - \lfloor \Delta N_{cm,il}^{TTI} / 2 \rfloor$$

For each transmission time interval of TrCH i with TF l , the rate-matching pattern is calculated with the algorithm in section 4.2.7.5. The following parameters are used as input:

$$X_i = N_{il}^{TTI} / 3N,$$

$$e_{ini} [1] = X_i; e_{ini} [2] = X_i \cdot e_{ini} = X_i,$$

$$e_{plus} [1] = 2 X_i; e_{plus} [2] = X_i \cdot e_{plus} = a \cdot X_i$$

$$e_{minus} [1] = 2 / \Delta N_i; e_{minus} [2] = 1 / \Delta N_i \cdot e_{minus} = a \cdot |\Delta N_i|$$

4.2.7.3 Bit separation and collection in uplink

The systematic bits (excluding bits for trellis termination) of turbo encoded TrCHs shall not be punctured. The systematic bit, first parity bit, and second parity bit in the bit sequence input to the rate matching block are therefore separated from each other. Puncturing is only applied to the parity bits and systematic bits used for trellis termination.

The bit separation function is transparent for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition. The bit separation and bit collection are illustrated in figures 6 and 7.

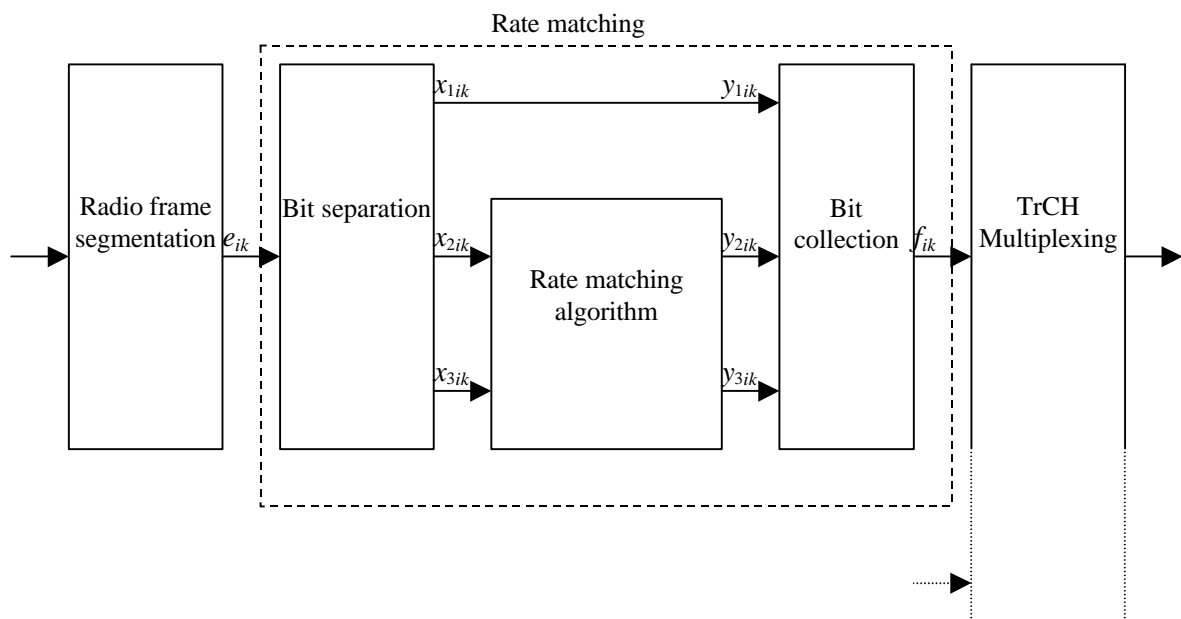


Figure 6: Puncturing of turbo encoded TrCHs

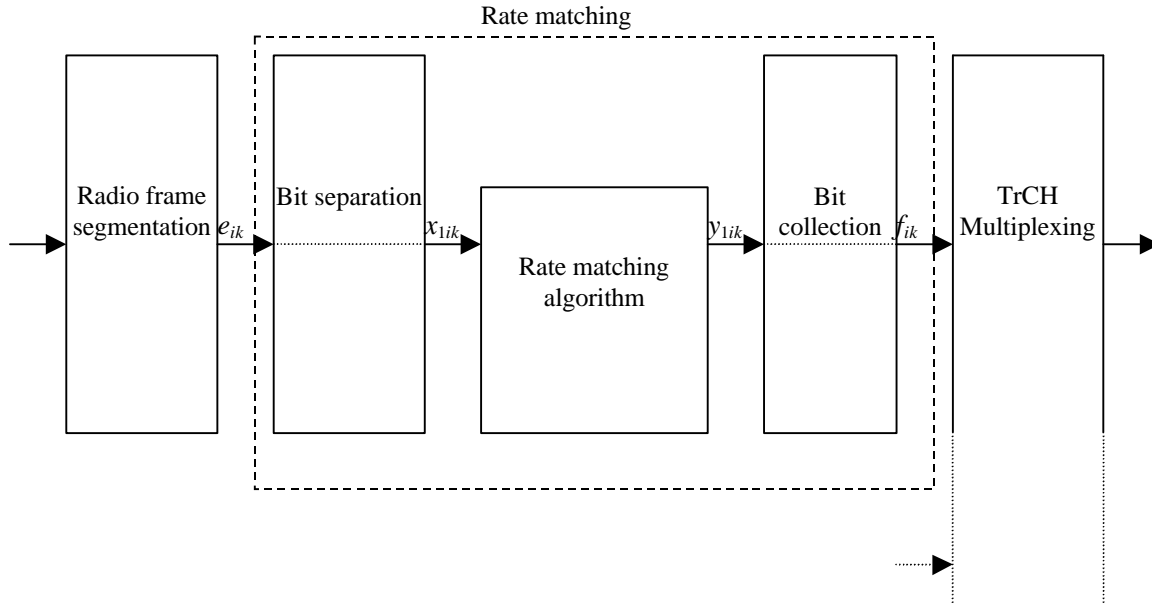


Figure 7: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition.

The bit separation is dependent on the 1st interleaving and offsets are used to define the separation for different TTIs. The offsets a_b for the systematic ($b=1$) and parity bits ($b \in \{2, 3\}$) are listed in table 4.

Table 4: TTI dependent offset needed for bit separation

TTI (ms)	a_1	a_2	a_3
10, 40	0	1	2
20, 80	0	2	1

The bit separation is different for different radio frames in the TTI. A second offset is therefore needed. The radio frame number for TrCH i is denoted by n_i , and the offset by b_{n_i} .

Table 5: Radio frame dependent offset needed for bit separation

TTI (ms)	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7
10	0	NA	NA	NA	NA	NA	NA	NA
20	0	1	NA	NA	NA	NA	NA	NA
40	0	1	2	0	NA	NA	NA	NA
80	0	1	2	0	1	2	0	1

4.2.7.3.1 Bit separation

The bits input to the rate matching are denoted by $e_{i1}, e_{i2}, e_{i3}, \dots, e_{iN_i}$, where i is the TrCH number and N_i is the number of bits input to the rate matching block. Note that the transport format combination number j for simplicity has been left out in the bit numbering, i.e. $N_i = N_{ij}$. The bits after separation are denoted by $x_{bi1}, x_{bi2}, x_{bi3}, \dots, x_{biX_i}$. For turbo encoded TrCHs with puncturing, b indicates systematic, first parity, or second parity bit. For all other cases b is defined to be 1. X_i is the number of bits in each separated bit sequence. The relation between e_{ik} and x_{bik} is given below.

For turbo encoded TrCHs with puncturing:

$$x_{1,i,k} = e_{i,3(k-1)+1+(a_1+b_{n_i}) \bmod 3} \quad k = 1, 2, 3, \dots, X_i \quad X_i = \lfloor N_i / 3 \rfloor$$

$$x_{1,i,\lfloor N_i/3 \rfloor+k} = e_{i,3\lfloor N_i/3 \rfloor+k} \quad k = 1, \dots, N_i \bmod 3 \quad \text{Note: When } (N_i \bmod 3) = 0 \text{ this row is not needed.}$$

$$x_{2,i,k} = e_{i,3(k-1)+1+(a_2+b_{n_i}) \bmod 3} \quad k = 1, 2, 3, \dots, X_i \quad X_i = \lfloor N_i/3 \rfloor$$

$$x_{3,i,k} = e_{i,3(k-1)+1+(a_3+b_{n_i}) \bmod 3} \quad k = 1, 2, 3, \dots, X_i \quad X_i = \lfloor N_i/3 \rfloor$$

For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

$$x_{1,i,k} = e_{i,k} \quad k = 1, 2, 3, \dots, X_i \quad X_i = N_i$$

4.2.7.3.2 Bit collection

The bits x_{bik} are input to the rate matching algorithm described in section 4.2.7.5. The bits output from the rate matching algorithm are denoted $y_1, y_2, y_3, \dots, y_{Y_i}$.

Bit collection is the inverse function of the separation. The bits after collection are denoted by $z_{bi1}, z_{bi2}, z_{bi3}, \dots, z_{biY_i}$.

After bit collection, the bits indicated as punctured are removed and the bits are then denoted by $f_{i1}, f_{i2}, f_{i3}, \dots, f_{iV_i}$, where i is the TrCH number and $V_i = N_i + DN_{ij}$. The relations between y_{bik} , z_{bik} , and f_{ik} are given below.

For turbo encoded TrCHs with puncturing ($Y_i = X_i$):

$$z_{i,3(k-1)+1+(a_1+b_{n_i}) \bmod 3} = y_{1,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

$$z_{i,3\lfloor N_i/3 \rfloor+k} = y_{1,i,\lfloor N_i/3 \rfloor+k} \quad k = 1, \dots, N_i \bmod 3 \quad \text{Note: When } (N_i \bmod 3) = 0 \text{ this row is not needed.}$$

$$z_{i,3(k-1)+1+(a_2+b_{n_i}) \bmod 3} = y_{2,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

$$z_{i,3(k-1)+1+(a_3+b_{n_i}) \bmod 3} = y_{3,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

After the bit collection, bits $z_{i,k}$ with value \mathbf{d} , where $\mathbf{d} \in \{0, 1\}$, are removed from the bit sequence. Bit $f_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index k after puncturing, bit $f_{i,2}$ corresponds to the bit $z_{i,k}$ with second smallest index k after puncturing, and so on.

For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

$$z_{i,k} = y_{1,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

When repetition is used, $f_{i,k} = z_{i,k}$ and $Y_i = V_i$.

When puncturing is used, $Y_i = X_i$ and bits $z_{i,k}$ with value \mathbf{d} , where $\mathbf{d} \in \{0, 1\}$, are removed from the bit sequence. Bit $f_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index k after puncturing, bit $f_{i,2}$ corresponds to the bit $z_{i,k}$ with second smallest index k after puncturing, and so on.

4.2.7.4 Bit separation and collection in downlink

The systematic bits (excluding bits for trellis termination) of turbo encoded TrCHs shall not be punctured. The systematic bit, first parity bit, and second parity bit in the bit sequence input to the rate matching block are therefore separated from each other. Puncturing is only applied to the parity bits and systematic bits used for trellis termination.

The bit separation function is transparent for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition. The bit separation and bit collection are illustrated in figures 8 and 9.

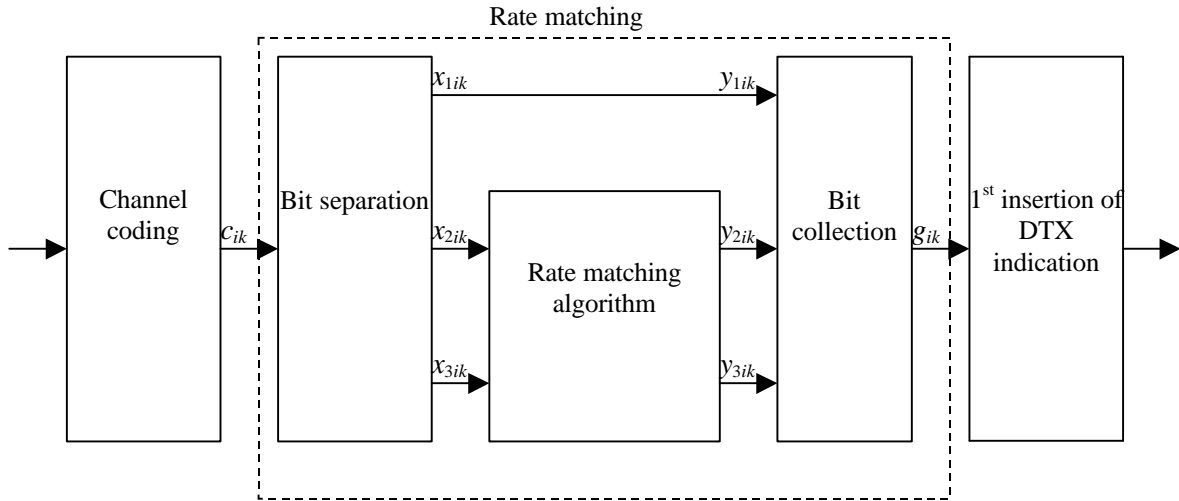


Figure 8: Puncturing of turbo encoded TrCHs

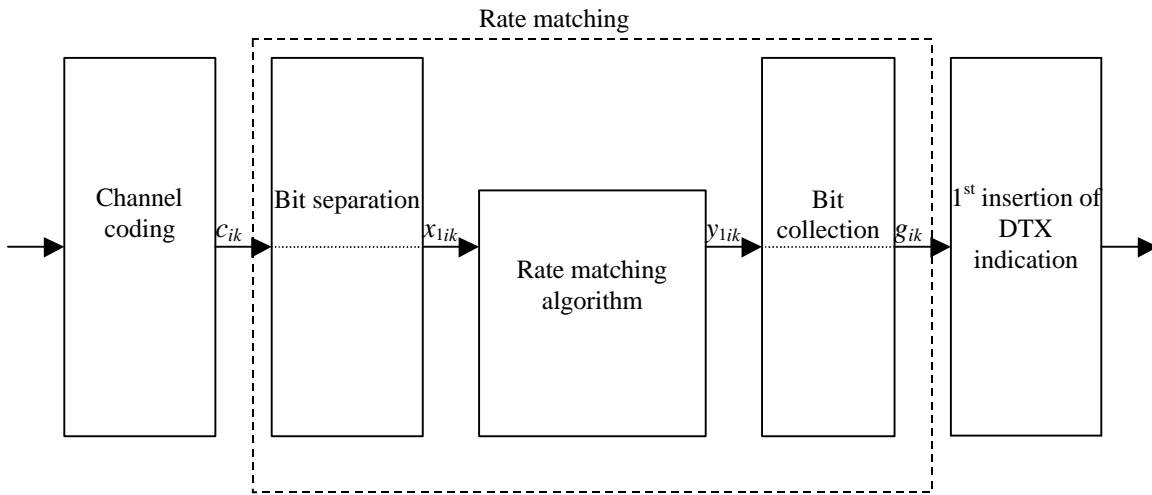


Figure 9: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition.

4.2.7.4.1 Bit separation

The bits input to the rate matching are denoted by $c_{i1}, c_{i2}, c_{i3}, \dots, c_{iE_i}$, where i is the TrCH number and E_i is the number of bits input to the rate matching block. Note that E_i is a multiple of 3 for turbo encoded TrCHs and that the transport format combination number j for simplicity has been left out in the bit numbering, i.e. $E_i = N_{ij}$. The bits after separation are denoted by $x_{bi1}, x_{bi2}, x_{bi3}, \dots, x_{biX_i}$. For turbo encoded TrCHs with puncturing, b indicates systematic, first parity, or second parity bit. For all other cases b is defined to be 1. X_i is the number of bits in each separated bit sequence. The relation between c_{ik} and x_{bik} is given below.

For turbo encoded TrCHs with puncturing:

$$x_{1,i,k} = c_{i,3(k-1)+1} \quad k = 1, 2, 3, \dots, X_i \quad X_i = E_i / 3$$

$$x_{2,i,k} = c_{i,3(k-1)+2} \quad k = 1, 2, 3, \dots, X_i \quad X_i = E_i / 3$$

$$x_{3,i,k} = c_{i,3(k-1)+3} \quad k = 1, 2, 3, \dots, X_i \quad X_i = E_i / 3$$

For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

$$x_{1,i,k} = c_{i,k} \quad k = 1, 2, 3, \dots, X_i \quad X_i = E_i$$

4.2.7.4.2 Bit collection

The bits x_{bik} are input to the rate matching algorithm described in section 4.2.7.5. The bits output from the rate matching algorithm are denoted $y_{bi1}, y_{bi2}, y_{bi3}, \dots, y_{biY_i}$.

Bit collection is the inverse function of the separation. The bits after collection are denoted by $z_{bi1}, z_{bi2}, z_{bi3}, \dots, z_{biY_i}$. After bit collection, the bits indicated as punctured are removed and the bits are then denoted by $g_{i1}, g_{i2}, g_{i3}, \dots, g_{iG_i}$, where i is the TrCH number and $G_i = N_{ij} + DN_{ij}$. The relations between y_{bik} , z_{bik} , and g_{ik} are given below.

For turbo encoded TrCHs with puncturing ($Y_i = X_i$):

$$z_{i,3(k-1)+1} = y_{1,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

$$z_{i,3(k-1)+2} = y_{2,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

$$z_{i,3(k-1)+3} = y_{3,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

After the bit collection, bits $z_{i,k}$ with value \mathbf{d} , where $\mathbf{d} \in \{0, 1\}$, are removed from the bit sequence. Bit $g_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index k after puncturing, bit $g_{i,2}$ corresponds to the bit $z_{i,k}$ with second smallest index k after puncturing, and so on.

For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

$$z_{i,k} = y_{1,i,k} \quad k = 1, 2, 3, \dots, Y_i$$

When repetition is used, $g_{i,k} = z_{i,k}$ and $Y_i = G_i$.

When puncturing is used, $Y_i = X_i$ and bits $z_{i,k}$ with value \mathbf{d} , where $\mathbf{d} \in \{0, 1\}$, are removed from the bit sequence. Bit $g_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index k after puncturing, bit $g_{i,2}$ corresponds to the bit $z_{i,k}$ with second smallest index k after puncturing, and so on.

4.2.7.5 Rate matching pattern determination

Referring to Figures 6,7,8 and 9, rate matching algorithm applies to flow of bits x_{1ik} for Convolutional Codes and Turbo Codes with repetition, and to flow of bits $x_{2i0}, x_{3i0}, x_{2i1}, x_{3i1}, \dots, x_{2iXi}, x_{3iXi}$ in case of Turbo Codes with puncturing.

Denote the bits before rate matching by:

$x_{i1}, x_{i2}, x_{i3}, \dots, x_{iX_i}, x_{i1}, x_{i2}, x_{i3}, \dots, x_{i\alpha X_i}$ where i is the TrCH number and X_i is the parameter given in sections 4.2.7.1 and 4.2.7.2, and $\alpha = 1$ for Convolutional Codes and Turbo Codes with repetition, $\alpha = 2$ for Turbo codes with puncturing.

The rate matching rule is as follows:

If Convolutional Codes then

$e[1] = e_{ini}[1]$ -- initial error between current and desired puncturing ratio

$\alpha = 1$ -- parameter for the range of index m

endif

if Turbo Codes then

$e[1] = e_{ini}[1]$ -- initial error between current and desired puncturing ratio for Parity 1 bits

$e[2] = e_{ini}[2]$ -- initial error between current and desired puncturing ratio for Parity 2 bits

if puncturing is needed then $\alpha = 2$ else $\alpha = 1$ endelse

endif

$m = 1$ index of current bit in the information bit flow (input flow)

$n = 1$ -- index of bit in the output flow before first interleaver

for $i = 0$ to $F_i - 1$ do $cbi[i] = 0$ -- initiate counter of number of bits inserted in each frame of the TTI

if puncturing is to be performed

$e = e_{ini}$ -- initial error between current and desired puncturing ratio

$m = 1$ -- index of current bit

do while $m \leq \alpha X_i$

if Turbo Codes then

$c = m \bmod 2$ - $c = 1$ if Parity-1 bit, $c = 2$ if Parity-2 bit

if $(c == 1)$ then $n = n + 1$ endif - in output flow increment output index to take into account bit X

else

$c = 1$

endelse

$e[c] = e[c] - e_{minus}[c]$ -- update error

if $e[c] \leq 0$ then -- check if bit number m should be punctured

set bit $x_{i,m}$ to d where $d \in \{0, 1\}$ $e[c] = e[c] + e_{plus}[c]$ -- update error

endif else

$col = BR[(n-1) \bmod F_i]$ -- calculate index of column where this bit will

go

while $cbi[col] < |\Delta N^{cm}[col]|$ do

insert one bit marked « p » in output position n -- insert bit marked to be removed in

compressed frame when creating the gap

$n = n + 1$ -- update output index since 1 bit is inserted

$cbi[col] = cbi[col] + 1$ -- update counter of bits p inserted in this column

$col = BR[(n-1) \bmod F_i]$ -- update column value of next position

enddo

$n = n + 1$ -- update output index, since bit x_m was not punctured

endelse

$m = m + 1$ -- next bit

```

end do

else
e = eini ----- initial error between current and desired puncturing ratio
m = 1 ----- index of current bit
do while m <=  $\alpha X_i$ 
    e = e - eminus      -- update error
    col = BR[(n-1) mod Fi]
    do while cbi[col] <  $\Delta N^{cm}$ [col]
    insert one bit marked p in position n
    n = n + 1 ----- update output index
    cbi[col] = cbi[col] + 1 ----- update counter of bits p inserted in this column
    col = BR[ (n-1) mod Fi] ----- update column value of next position
enddo
do while e <= 0      -- check if bit number m should be repeated
    col = BR[ (n-1) mod Fi] ----- calculate column where the current position will go
    do while cbi[col] <  $\Delta N^{cm}$ [col]
    insert one bit marked p
    n = n + 1 ----- update output index since 1 bit has been inserted
    cbi[col] = cbi[col] + 1 ----- update counter of bits p inserted in this column
    col = BR[ (n-1) mod Fi] ----- update column value of next position
enddo
    repeat bit  $x_{i,m}$ 
    n = n + 1 ----- update output index since bit has been repeated
    e = e + eplus -- update error
end do
m = m + 1      -- next bit
end do
end if

```

A repeated bit is placed directly after the original one.

Table 10 shows the detailed parameters for each transmission gap length for the different transmission time reduction methods.